LECTURE 10: GNC + LOW-LEVEL VISION

[•] GRADUATED NON-CONVEXITY (GNC)

Problem: Robust objective functions are not convex

e.g. Lorentzian:

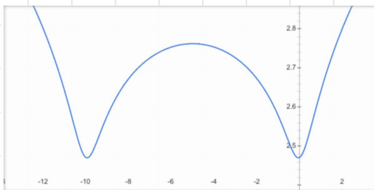
$$f(x) = \log\left(1 + \frac{1}{2}\left(\frac{x}{\sigma}\right)^2\right)$$

Charbonnier:

$$f(x) = (x^2 + \varepsilon^2)^a \quad ; \; a < 0.5$$

$a = 0.5 \rightarrow$ convex & differentiable.

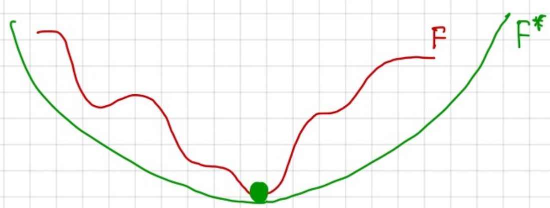
Non-convex: Many local minima cause the optimization to be trapped.



e.g: $\left[(x+10)^2 + \varepsilon^2\right]^{0.1} + \left[x^2 + \varepsilon^2\right]^{0.1}$

Solution: Graduate Non-Convexity (GNC)

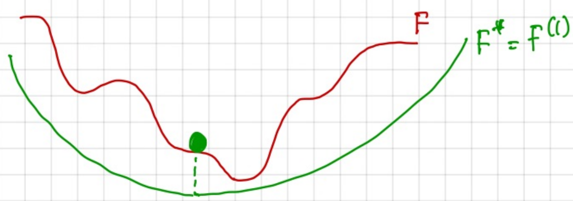
Ideal case:



The approximation function F^* provides the correct minimum.

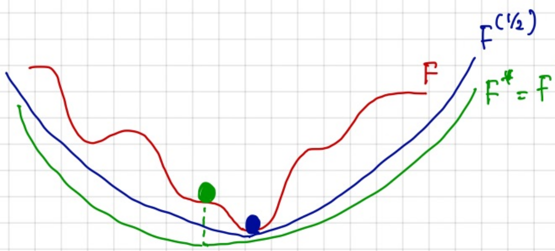
Practically:

(1)



The approximation doesn't directly indicate the minimum but the approximated function $F^{(1)}$ is convex and generates the initial value of Δu.

(2)



Since $F^{(1/2)}$ gets closer to the actual function F, and is convex, it can generate a better value of Δu.

GNC for optical flow:

$$E_c(u,v) = \lambda E_Q(u,v) + (1-\lambda)\underbrace{E(u,v)}_{\text{Original non-convex function}}$$

where:

- $E(u,v) = f(I_x u + I_y v + I_t) + \alpha\left[f(\nabla u) + f(\nabla v)\right]$

  $f(x)$ is a non-convex function, yet differentiable

  e.g.: Charbonnier with $a < 0.5$

- $E_Q(u,v) = (I_x u + I_y v + I_t)^2 + \alpha\left(\|\nabla u\|_2^2 + \|\nabla v\|_2^2\right)$

  ↳ Convex function.

$$\boxed{J(u,v) = \iint \left[\lambda E_Q(u,v) + (1-\lambda) E(u,v)\right] dx\, dy}$$

- λ is 1 in the beginning of the iteration, producing $(u,v)_1$
- λ is set to 0.5 in the second iteration, the initial $(u,v)_{init} = (u,v)_1$ producing $(u,v)_2$
- λ is set to 0 in the third iteration, with initial $(u,v)_2$, & producing $(u,v)_{final}$.

Q: Why don't we directly solve: $J(u,v) = \int E(u,v)\, dx\, dy$?

A: Our solution will be trapped in a local minimum → non convex.
However, if we initialize step-by-step, we can avoid local minima.

[•] STRUCTURE - TEXTURE DECOMPOSITION

Rudin - Osher - Fatemi (ROF) algorithm :

Model : $I = I_s + I_t$

Goal :

I Input | I_s Structure Layer | I_t Texture Layer

Objective function :

$$I_s^* = \underset{\{I_s\}}{\operatorname{argmin}} \sum_x \left(I_s(x) - I(x) \right)^2 + \lambda \, |\nabla I_s(x)|_2$$

where : $$|\nabla I_s(x)|_2 = \sqrt{\left(\frac{\partial}{\partial x} I_s\right)^2 + \left(\frac{\partial}{\partial y} I_s\right)^2} = \sqrt{I_{s_x}^2 + I_{s_y}^2}$$

Or : $$I_s = \underset{\{I_s\}}{\operatorname{argmin}} \; J(I_s)$$

where : $$J(I_s) = \iint (I_s(x) - I(x))^2 + \lambda \, |\nabla I_s|_2 \; dx\,dy$$

$$\iint |\nabla I_s|_2 \; dx\,dy \longrightarrow$$ L2 - Total Variation (L2-TV)

Q: How to minimize $J(I_s)$?

A: ROF's "Non-linear total variation based noise removal alg.", 1992.

$$E(I_s) = (I_s(x) - I(x))^2 + \lambda \underbrace{|\nabla I_s(x)|_2}_{= \sqrt{I_{sx}^2 + I_{sy}^2}}$$

$$J(I_s) = \iint E(I_s)\, dx\, dy$$

Using the Euler-Lagrange equation:

$$\nabla J = \frac{\partial E}{\partial I_s} - \frac{\partial}{\partial x} \frac{\partial E}{\partial I_{sx}} - \frac{\partial}{\partial y} \frac{\partial E}{\partial I_{sy}} = 0$$

$$\frac{\partial E}{\partial I_s} = 2(I_s(x) - I(x)) \; ; \qquad \frac{\partial E}{\partial I_{sx}} = \frac{\lambda}{2} \frac{2 I_{sx}}{\sqrt{I_{sx}^2 + I_{sy}^2}}$$

$$\nabla J = 2(I_s(x) - I(x)) - \frac{\partial}{\partial x} \frac{\lambda I_{sx}(x)}{\sqrt{I_{sx}^2(x) + I_{sy}^2(x)}} - \frac{\partial}{\partial y} \frac{\lambda I_{sy}(x)}{\sqrt{I_{sx}^2(x) + I_{sy}^2(x)}} = 0$$

Gradient descent:

$$I_s^{new} = I_s^{old} - \alpha \nabla J(I_s) \Big|_{I_s = I_s^{old}}$$

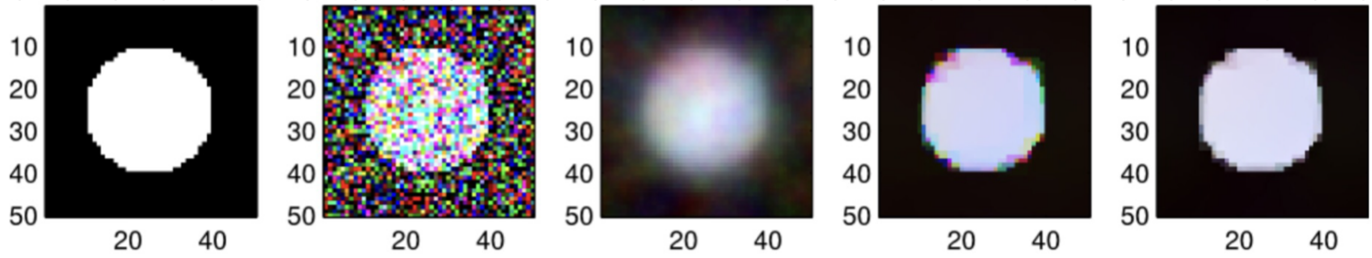
$$= I_{s(x)}^{old} - \alpha \left[2(I_{s(x)}^{old} - I(x)) - \frac{\partial}{\partial x} \frac{\lambda I_{sx}^{old}(x)}{\sqrt{I_{sx}^2(x) + I_{sy}^2(x)}} - \frac{\partial}{\partial y} \frac{\lambda I_{sy}^{old}(x)}{\sqrt{I_{sx}^2(x) + I_{sy}^2(x)}} \right]$$

$$= (1 - 2\alpha) I_s^{old} + 2\alpha I + \frac{\partial}{\partial x} \frac{\lambda I_{sx}^{old}}{\sqrt{I_{sx}^2 + I_{sy}^2}} + \frac{\partial}{\partial y} \frac{\lambda I_{sy}^{old}}{\sqrt{I_{sx}^2 + I_{sy}^2}}$$

Initialization: $I_s^{init} = I$

Q: Why don't we use $\|\nabla I_s(\bar{x})\|_2^2$, which is easy to optimize?

A: This quadratic L_2-norm generates blurry outputs, while the TV prior generate sharper edges.

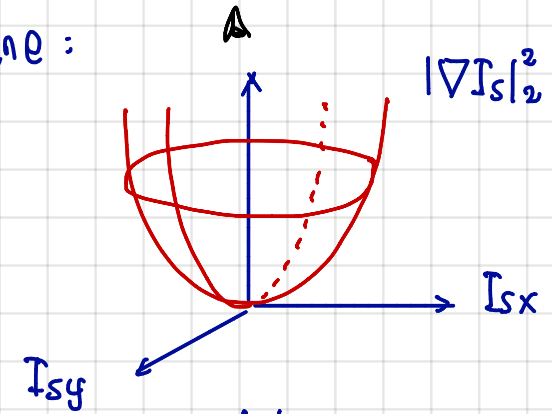


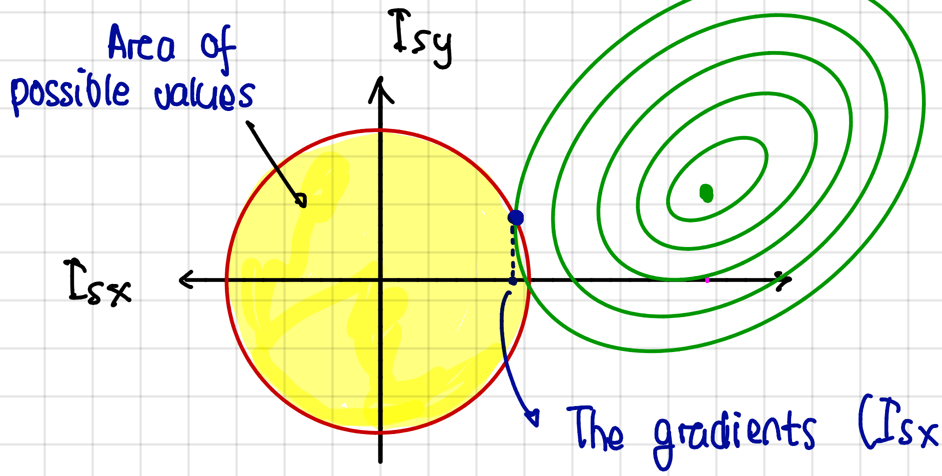
clean image | noisy image (input) | Quadratic L_2-norm | L1-TV | L2-TV

Q: Why does the quadratic L2-norm regularization cause the recovered image to be blurry?

A: Quadratic L2 norm: $|\nabla I_s|_2^2 = I_{sx}^2 + I_{sy}^2$

Imagine:

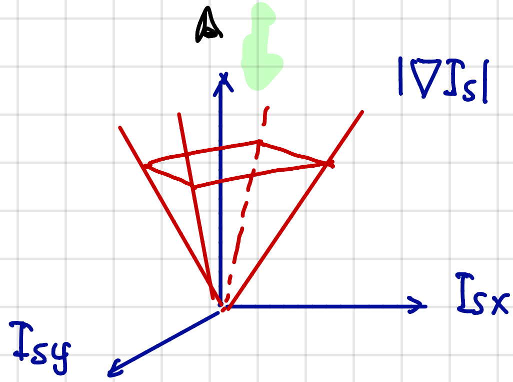


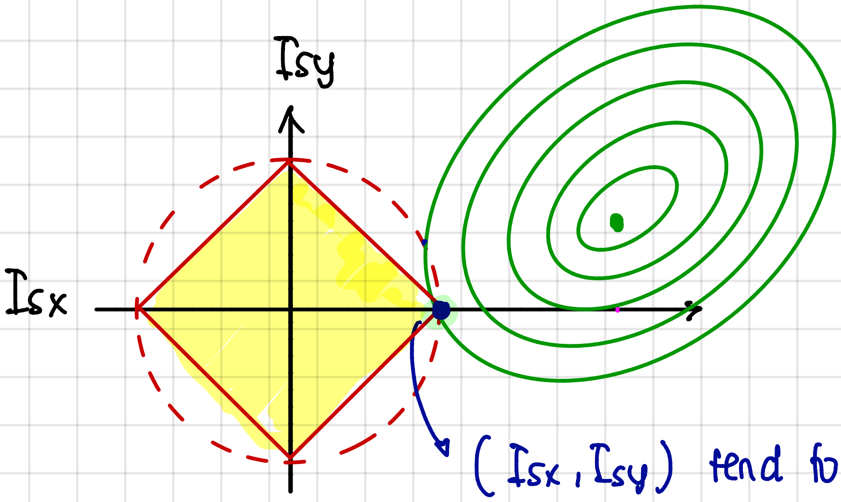


constraint from the data term (which might not be convex)

The gradients (I_{sx}, I_{sy}) are likely not zero.

L1-norm:





$f(x) = x^p$

Note: If (I_{sx}, I_{sy}) are mostly zero (sparse), the values of I_s will be sharper.

L2-TV $(|\nabla I_s|_2)$ is not exactly L1 norm, but sparser than that of L2-norm $(|\nabla I_s|_2^2)$.